Towards scalable fuzzy–rough feature selection

Jensen, Richard; MacParthalain, Neil

Published in:
Information Sciences

DOI:
10.1016/j.ins.2015.06.025

Publication date:
2015

Citation for published version (APA):
Jensen, R., & MacParthalain, N. (2015). Towards scalable fuzzy–rough feature selection. *Information Sciences*, *323*, 1-15. https://doi.org/10.1016/j.ins.2015.06.025

Towards Scalable Fuzzy-Rough Feature Selection

Richard Jensen, Neil Mac Parthaláin*

Department of Computer Science, Aberystwyth University, Aberystwyth, Ceredigion, SY23 3DB, Wales, UK.
(email:{rkj,ncm}@aber.ac.uk)

Abstract

Research in the area of fuzzy-rough set theory, and its application to feature or attribute selection in particular, has enjoyed much attention in recent years. Indeed, with the growth of larger and larger data dimensionality, the number of data objects required in order to generate accurate models increases exponentially. Thus, for model learning, feature selection has become increasingly necessary. The use of fuzzy-rough sets as dataset pre-processors offer much in the way of flexibility, however the underlying complexity of the subset evaluation metric often presents a problem and can result in a great deal of potentially unnecessary computational effort. This paper proposes two different novel ways to address this problem using a neighbourhood approximation step and attribute grouping in order to alleviate the processing overhead and reduce complexity. A series of experiments are conducted on benchmark datasets which demonstrate that much computational effort can be avoided, and as a result the efficiency of the feature selection process for fuzzy-rough sets can be improved considerably.
Keywords: Fuzzy-Rough Sets, Feature Selection, Nearest Neighbours, Feature Grouping.

1. Introduction

The unrelenting surge in the growth of data dimensionality in recent times has had the effect of highlighting the weaknesses of many of the traditional

*Corresponding author

feature selection (FS) methods in terms of their scalability. Although there have been some efforts to address the problem of larger data dimensionality [9], [25], the overall response from computational intelligence researchers has been to adapt existing techniques for distributed processing environments using Hadoop and MapReduce [2], [3], [24], [26] rather than attempting to tackle the problem more directly.

The use of fuzzy-rough set theory (FRS) for the task of feature selection has proven remarkably popular in recent years. Indeed the theory has been the subject of numerous modifications and extensions [5], [13], [22]. However, despite these extensions, even the most basic interpretations of FRS suffer in terms of their ability to scale to large data. One of the drawbacks is related to the way in which the FRS lower approximation is defined and computed. The problem is related to the fact that all of the objects in the data must be considered when calculating the membership to the lower approximation. Indeed, the subset evaluation metric relies upon these calculations and is thus directly affected by this situation. Clearly, these issues become an obstacle to dealing with data, particularly when the data is large. However, consider the case where both the number of data objects is large *and* the dimensionality is also large. In this particular case, the problem is compounded even further meaning that approaches based on FRS suffer from a computational overhead that clearly becomes prohibitive.

Fuzzy-rough set theory extends the rough set approximation operators by fuzzifying the indiscernibility relation as well as the concept itself. This generalisation provides much greater flexibility, however, the most commonly utilised definitions of fuzzy-rough approximations ignore some important aspects. In traditional fuzzy-rough sets, all data objects in the dataset must be considered when generating the approximations used in the fuzzy-rough dependency calculation. This means that considerable computational effort is expended each time the lower approximation memberships are calculated. For feature selection, this occurs with the consideration of each candidate subset, meaning that a large number of membership calculations are made. In addition, even small

changes in the data distribution can often mean that the generated approximations can vary greatly. This can also have a negative impact on the stability of any technique based upon such definitions.

In an attempt to alleviate the aforementioned problems, two alternative approaches to improving fuzzy-rough FS are presented here. The first approach works by reformulating the way in which membership degrees to the approximations are computed by including only those data objects which are k-nearest neighbours and are also *not* of the same decision class as the data object under consideration. This reduces the impact of the *number of objects* in data on FRS-based methods. The second approach offers a form of grouping and ranking of the features which are then framed in the context of a modified search, with features drawn from groups rather than from the full set of features. This reduces the impact of the *number of features*, and is applicable to other feature selection methods. These techniques offer a starting point for further development in terms of improving the scalability of fuzzy-rough approaches for FS.

The remainder of the paper is structured as follows: the preliminaries for fuzzy-rough set theory and feature selection are covered in Section 2 along with an in-depth examination of the factors which affect the complexity of fuzzy-rough feature selection in the presence of large data. Section 3 presents the first of two different approaches to tackling these problems: nearest neighbour-based fuzzy-rough sets. Section 4 presents the fuzzy-rough feature grouping approach. An experimental evaluation is carried out in Section 5 where both approaches are applied to a number of different datasets. Finally, Section 6 concludes the paper, with some discussion and identification of a number of potential areas for future development.

2. Theoretical Background

One of the main problems associated with large dimensionality, means that any attempt to use machine learning tools to extract knowledge, results in very poor performance. Feature selection (FS) is a process which attempts to select

features which are information-rich and also retain the original meaning of the features following reduction. The search for feature subsets is performed in a combinatorially large space and thus presents a major challenge for data mining approaches.

Let $I = (\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a non-empty set of finite objects (the universe of discourse) and $\mathbb{A}$ is a non-empty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a,\ \forall a \in \mathbb{A}\ .\ V_a$ is the set of values that attribute a may take. For a decision system, $\mathbb{A} = \{\mathbb{C} \cup \mathbb{D}\}$ where $\mathbb{C}$ is the set of input features and $\mathbb{D}$ is the set of class or decision indices.

For any $P \subseteq \mathbb{C}$, there exists an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in \mathbb{U}^2 | \forall a \in P, a(x) = a(y)\} \tag{1}$$

The partition generated by $IND(P)$ is denoted $\mathbb{U}/P$ and is calculated as follows:

$$\mathbb{U}/P = \otimes\{\mathbb{U}/IND(\{a\}) : a \in P\} \tag{2}$$

where,

$$S \otimes T = \{X \cap Y : \forall X \in S, \forall Y \in T, X \cap Y \neq \emptyset\} \tag{3}$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. Let $X \subseteq \mathbb{U}$. X can be approximated using only the information contained in P by constructing the P-*lower* and P-*upper* approximations of X:

$$\underline{P}X = \{x : [x]_P \subseteq X\} \tag{4}$$

$$\overline{P}X = \{x : [x]_P \cap X \neq \emptyset\} \tag{5}$$

In the original work of [21], the lower approximation of a set X is constructed using a subset of the conditional attributes $P \subseteq \mathbb{C}$ w.r.t. a crisp equivalence relation. The positive region can then be generated, which contains those data objects in the universe $\mathbb{U}$ for which the values of P allow to predict the decision

classes in $\mathbb{D}$ unequivocally: $POS_P(\mathbb{D}) = \bigcup_{X \in \mathbb{U}/\mathbb{D}} \underline{P}X$. Based on the positive region, the rough set degree of dependency of the decision attribute(s) $\mathbb{D}$ on a set of attributes P can be calculated: $\gamma_P(\mathbb{D}) = \frac{|POS_P(\mathbb{D})|}{|\mathbb{U}|}$. This measure can then be used to gauge subset quality for (crisp) rough set-based FS.

2.1. Fuzzy-Rough Sets

A fuzzy-rough set [8] is defined by two fuzzy sets, fuzzy lower and upper approximations, obtained by extending the corresponding crisp rough set notions defined previously in (4) and (5).

In the crisp case, elements that belong to the lower approximation (i.e. have a membership of 1.0) are said to belong to the approximated set with absolute certainty. In the fuzzy-rough case, elements may have a membership in the range [0,1], thus allowing greater flexibility in modelling uncertainty. Definitions for the fuzzy lower and upper approximations can be found in [23]. For the work described here, only the fuzzy lower approximation is utilised, where a fuzzy indiscernibility relation is used to approximate a fuzzy concept X:

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} \mathcal{I}(\mu_{R_P}(x,y), \mu_X(y)) \tag{6}$$

where $\mathcal{I}$ is a fuzzy implicator. A fuzzy implicator is any $[0,1]^2 \to [0,1]$ mapping that is decreasing in its first and increasing in its second argument, which satisfies $\mathcal{I}(0,0) = \mathcal{I}(0,1) = \mathcal{I}(1,1) = 1$ and $\mathcal{I}(1,0) = 0$. R_P is the fuzzy similarity relation induced by the subset of features P:

$$\mu_{R_P}(x,y) = \mathcal{T}_{a \in P}\{\mu_{R_a}(x,y)\} \tag{7}$$

where $\mu_{R_a}(x,y)$ is the degree to which objects x and y are similar for feature a, and may be defined in many ways [23], and $\mathcal{T}$ is a t-norm, an increasing, commutative, associative $[0,1]^2 \to [0,1]$ mapping satisfying $\mathcal{T}(x,1) = x$ for x in $[0,1]$. In a similar way to the original crisp rough set approach, the fuzzy positive region [17] can be defined as:

$$\mu_{POS_P(\mathbb{D})}(x) = \sup_{X \in \mathbb{U}/\mathbb{D}} \mu_{\underline{R_P}X}(x) \tag{8}$$

An important issue in data analysis is the discovery of dependencies between features. The fuzzy-rough degree of dependency of $\mathbb{D}$ on the attribute subset P can be defined in the following way:

$$\gamma'_P(\mathbb{D}) = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P(\mathbb{D})}(x)}{|\mathbb{U}|} \tag{9}$$

A fuzzy-rough reduct *Red* is a minimal subset of features (i.e. there is no redundancy) that preserves the dependency degree of the entire dataset, i.e. $\gamma'_{Red}(\mathbb{D}) = \gamma'_{\mathbb{C}}(\mathbb{D})$. Based on this, subset search techniques can be used that employ equation (9) in order to gauge subset quality.

2.2. Fuzzy Discernibility

Crisp discernibility matrices, often used in rough set feature selection, may also be extended for use in fuzzy-rough feature selection [17]. The entries (known as clauses) in a fuzzy discernibility matrix (FDM) are a fuzzy set, to which every feature belongs to a certain degree. The extent to which a feature a belongs to the fuzzy clause C_{ij} is determined by the following:

$$\mu_{C_{ij}}(a) = N(\mu_{R_a}(i, j)) \tag{10}$$

where N denotes fuzzy negation and $\mu_{R_a}(i, j)$ is the fuzzy similarity of objects i and j, and hence $\mu_{C_{ij}}(a)$ is a measure of the fuzzy discernibility. For the crisp case, if $\mu_{C_{ij}}(a) = 1$ then the two objects are distinct for this feature; if $\mu_{C_{ij}}(a) = 0$, the two objects are identical. For fuzzy cases where $\mu_{C_{ij}}(a) \in (0, 1)$, the objects are partly discernible. Each entry (or clause) in the fuzzy indiscernibility matrix is a set of attributes and their memberships:

$$C_{ij} = \{a_x | a \in \mathbb{C}, x = N(\mu_{R_a}(i, j))\}\ i, j = 1, ..., |\mathbb{U}| \tag{11}$$

For example, an entry C_{ij} in the fuzzy discernibility matrix may be:
$\{a_{0.4}, b_{0.8}, c_{0.2}, d_{0.0}\}$. This denotes that $\mu_{C_{ij}}(a) = 0.4$, $\mu_{C_{ij}}(b) = 0.8$, etc. In crisp discernibility matrices, these values are either 0 or 1 as the underlying relation is an equivalence relation. The example clause can be viewed as an indicator of the significance value for each feature - the extent to which the feature discriminates between objects i and j.

2.3. Fuzzy Discernibility Function

As with the crisp approach, entries in the matrix can be used to construct a fuzzy discernibility function:

$$f_D(a_1^*, ..., a_m^*) = \wedge\{\vee C_{ij}^* | 1 \leq j < i \leq |\mathbb{U}|\} \tag{12}$$

where $C_{ij}^* = \{a_x^* | a_x \in C_{ij}\}$. The function returns values in $[0, 1]$, which can be viewed as a measure of the extent to which the function is satisfied for a given assignment of truth values to variables. To discover reducts from the fuzzy discernibility function, the task is to find the minimal assignment of the value `true` to the variables such that the formula is maximally satisfied. By setting all variables to `true`, the maximal value for the function can be obtained as this provides the greatest discernibility between objects.

2.4. Decision-relative Fuzzy Discernibility Matrix

For a decision system, the decision feature must be taken into account in order to achieve valid reductions; only those clauses of a decision value which is different to that of the object under consideration are included in the matrix when generating any subsequent reduction. For the fuzzy version, this is encoded as:

$$f_D(a_1^*, ..., a_m^*) = \{\wedge\{\{\vee C_{ij}^*\} \leftarrow q_{N(\mu_{R_q}(i,j))}\} | 1 \leq j < i \leq |\mathbb{U}|\} \tag{13}$$

where $C_{ij}^* = \{a_x^* | a_x \in C_{ij}\}$, for decision feature q, where $\leftarrow$ denotes fuzzy implication. If $\mu_{C_{ij}}(q) = 1$ then this clause provides maximum discernibility (i.e., the two objects are maximally different according to the fuzzy similarity measure). When the decision is crisp and crisp equivalence is used, $\mu_{C_{ij}}(q)$ becomes either 0 or 1. The degree of satisfaction for a clause C_{ij} for a given subset of features P is defined as:

$$SAT_P(C_{ij}) = \mathcal{S}_{a \in P}\{\mu_{C_{ij}}(a)\} \tag{14}$$

for a t-conorm $\mathcal{S}$. In traditional (crisp) propositional satisfiability, a clause is fully satisfied if at least one variable in the clause has been set to `true`. For

the fuzzy case, clauses may be satisfied to a certain degree depending on which variables have been assigned the value `true`. By setting $P = \mathbb{C}$, the maximum degree of satisfiability for a clause can be obtained:

$$maxSAT_{ij} = SAT_{\mathbb{C}}(C_{ij}) = \mathcal{S}_{a \in \mathbb{C}}\{\mu_{C_{ij}}(a)\} \tag{15}$$

In this setting, a fuzzy-rough reduct corresponds to a (minimal) truth assignment to variables such that each clause has been satisfied to its maximal extent.

2.5. Complexity Aspects of Fuzzy-Rough Feature Selection

Feature selection approaches based upon fuzzy-rough sets have proven very popular. However, there are two particular aspects which present scalability problems for large data. The first relates to the number of data objects contained in the data, as the pairwise comparison of each data object with every other object in generating the fuzzy similarity relations means that this is a $O(n^2)$ operation (where n is the number of data objects). Also, the calculation of the dependency measure itself requires $O(n^2)$ comparisons. It is clear therefore, that an increase in the number of objects will have a negative effect upon the runtime of such approaches.

It has been shown in [13] and [22] that the standard approach to fuzzy-rough sets uses *only* the membership of the nearest data object that is of a different class to that of the objects under consideration. Therefore, there is much wasted computational effort. Recall the earlier definition of the fuzzy lower approximation:

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} \mathcal{I}(\mu_{R_P}(x, y), \mu_X(y))$$

Due to a natural property of fuzzy implicators and their use for calculating membership degrees; when the second component ($\mu_X(y)$) is 1.0 (i.e. `true`) then the implication result will evaluate to 1.0. This component corresponds to the degree to which an object belongs to a given decision class; a value of 1.0 indicates that the object is of the same decision class. Therefore, the only

data objects to have an impact upon the result of the implication operation are those of classes other than that of the object under consideration. Of these, the nearest object of a different class will produce the smallest value for the implication operation, and therefore, it is this value only that is used, due to the fact that above definition results in the minimum of *all* implications. The process which considers all neighbours is naturally very time-consuming and is exacerbated further when the data contains a large number of data objects. For feature selection (FS), it will therefore require the calculation of the nearest neighbours for *each* feature subset candidate that is considered by the selection algorithm. Hence, there is very little saving in time when employing such a nearest neighbour approach. The first approach presented in this paper, seeks to approximate the nearest neighbour calculations by computing the nearest neighbour(s) for each data object *prior* to computing the lower approximation. Although the final subsets produced may not be true reducts (in the fuzzy-rough sense), their computation will be much less intensive and thus methods based on this framework are applicable to larger data [19].

The second aspect is that of dimensionality. When dimensionality is large, then typical approaches to search which have been used traditionally (e.g. hill-climbing) can suffer from poor performance due to the combinatorially large space in which the search must be performed. The approach presented here focuses upon alleviating this overhead. It does this by a process of grouping the features prior to applying a modified hill-climbing search to the problem. It uses a correlation measure to determine the redundancy (or similarity) of the features prior to grouping them [20]. Correlation of each feature with respect to the class label is then used as an internal ranking within each group. It is of note that this approach is not limited to the use of fuzzy-rough evaluation metrics and any subset evaluation metric may be employed for selecting features.

The work presented in this paper therefore attempts to address these two problems by focusing upon each one individually. The result is two different approaches; one which uses a neighbourhood approximation for constructing approximations and one which groups the features prior to selection.

3. Nearest Neighbour-based Fuzzy-Rough Sets

As discussed previously, it has been shown in [13] and [22] that the standard approach to fuzzy-rough sets uses only the nearest data object of a different class when considering the membership of a data object to the lower approximation. Therefore, the only data objects to have an impact on the result of the implications are those of classes other than that of the object under consideration. Of these, the nearest object of a different class will produce the smallest value for the implication operation, and therefore, it is this value only that is used, due to the fact that equation (6) results in the minimum of all implications. This process (as mentioned previously), is quite time-consuming, as it requires the calculation of the nearest neighbours for each feature subset candidate that is considered. The approach presented here calculates the neighbours beforehand and uses only these neighbours in the evaluations of subsets.

3.1. nnFRFS

Using the approach described above, the original FRFS method can be altered to only consider the nearest neighbours, termed nnFRFS hereafter. The lower approximation is thus defined, for fuzzy concept X, feature subset P and fuzzy implicator $\mathcal{I}$:

$$\mu_{\underline{R_P^k}X}(x) = \inf_{y \in NN_x^k} \mathcal{I}(\mu_{R_P}(x,y), 0) \tag{16}$$

Each neighbour in NN_x^k has been determined *beforehand* using $R_{\mathbb{C}}$ to measure similarity and only considering those k nearest objects that belong to a different class than x. Those features present in the subset P are used for determining the similarity R_P. For standard nnFRFS, only the closest neighbour is required, so $|NN_x^1| = 1$ for all x, reducing the number of calculations drastically. This framework can be used for other extensions (such as VQRS and OWA-based fuzzy-rough feature selectors); for these, all neighbours will have some impact on the final calculation and so parameter k needs to be set appropriately.

In order to demonstrate that the parameter k has no impact on nnFRFS: assume that an object x has k neighbours. The fuzzy lower approximation using

these is $\inf_{y \in NN_x^k} \mathcal{I}(\mu_{R_P}(x,y),0)$, and hence the smallest implication evaluation will be the resultant membership of x to the lower approximation. This will always be the result of using the largest value for $\mu_{R_P}(x,y)$ due to the property of implicators, which is generated by considering the closest neighbour to x. In other words, as the closest neighbour of x always determines the lower approximation membership, the parameter k therefore has no impact.

Using the nearest neighbour-based fuzzy lower approximation, the fuzzy positive region can be redefined as:

$$\mu_{POS_P^k(\mathbb{D})}(x) = \sup_{X \in \mathbb{U}/\mathbb{D}} \mu_{\underline{R_P^k} X}(x) \tag{17}$$

The fuzzy-rough degree of dependency of $\mathbb{D}$ on the attribute subset P can then be redefined:

$$\gamma_P^k(\mathbb{D}) = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P^k(\mathbb{D})}(x)}{|\mathbb{U}|} \tag{18}$$

or, the normalised version (as the data may be inconsistent):

$$\gamma_P^k(\mathbb{D}) = \frac{1}{\mathbb{U}} \sum_{x \in \mathbb{U}} \frac{\mu_{POS_P^k(\mathbb{D})}(x)}{\mu_{POS_C^k(\mathbb{D})}(x)} \tag{19}$$

This measure of dependency can be used in the same way as the original definition as a basis for guiding search toward optimal subsets. In this paper, a greedy hill-climbing search method is used and implemented as shown in Fig. 1.

3.2. nnFDM

The fuzzy discernibility matrix-based approach described earlier can also be altered to form a more computationally-efficient process. Recall that the discernibility matrix is constructed by the pairwise comparison of all objects in a dataset, and for the decision-relative discernibility matrix, clauses are only generated when pairs of objects belong to different decision classes. Conditional features that differ in value between object pairs are recorded in the clauses; a subset of features then is required such that all clauses are satisfied, meaning that all objects can be discerned. For the fuzzy-rough approach, the importance

nnFRFS($\mathbb{C}$,$\mathbb{D}$,k).
$\mathbb{C}$, the set of all conditional attributes;
$\mathbb{D}$, the set of decision attributes;
k, the number of nearest neighbours to consider.

(1) $R \leftarrow \{\}$; $\gamma^k_{best} = 0$;
(2) **foreach** $x \in \mathbb{U}$, calculate NN^k_x
(3) **do**
(4) $T \leftarrow R$
(5) **foreach** $x \in (\mathbb{C} - R)$
(6) **if** $\gamma^k_{R\cup\{x\}}(\mathbb{D}) > \gamma^k_T(\mathbb{D})$
(7) $T \leftarrow R \cup \{x\}$
(8) $\gamma^k_{best} = \gamma^k_T(\mathbb{D})$
(9) $R \leftarrow T$
(10) **until** $\gamma^k_{best} == \gamma^k_{\mathbb{C}}(\mathbb{D})$
(11) **return** R

Figure 1: The nnFRFS algorithm

of features for a pair of objects is determined by the negation of the fuzzy similarity. Pairs of objects which are very similar but belong to different decision classes are therefore problematic, and the features that differ the most in value between them are very important.

The most important clauses for an object are those that are generated by the nearest neighbours of a different class. As more dissimilar objects are considered, the more features will appear in the clauses (or will belong to a higher degree), meaning that the clause is more easily satisfiable. Hence, the most useful information is contained in the nearest few neighbours for each object, as these are the most difficult to discern. The modified FDM approach presented here attempts to approximate the full set of clauses by only considering the most important clauses, generated by nearest neighbours of objects of different classes. The parameter k determines how many of the nearest objects are used to generate such clauses. Setting k to $|\mathbb{U}| - 1$ will produce all possible clauses, and the algorithm will collapse to the original FDM approach.

Each entry in the fuzzy discernibility matrix is generated by comparing pairs of objects. Here, only the k nearest objects of a different class are considered.

Clauses are generated in the same way as for the fuzzy discernibility matrix approach described previously. Based on this, the full set of clauses can be generated as follows:

$$Clauses^k = \{C_{ij} |\ j \in NN_i^k \vee i \in NN_j^k\} \tag{20}$$

where NN_i^k is the set of k nearest neighbours for object i, generated in the same way as for nnFRFS previously. Therefore, a clause is generated from object pair i, j if at least one of the objects appears in the other's nearest neighbour list.

The degree of satisfaction of a clause C for a subset of features P is defined as:

$$SAT_P(C) = \mathcal{S}_{a \in P}\{\mu_C(a)\} \tag{21}$$

for t-conorm $\mathcal{S}$. By setting $P = \mathbb{C}$, the maximum satisfiability degree of a clause C can be obtained:

$$maxSAT_C = SAT_{\mathbb{C}}(C) = \mathcal{S}_{a \in \mathbb{C}}\{\mu_C(a)\} \tag{22}$$

Finally, the following subset evaluation measure can be used to gauge the worth of a subset of features P:

$$\tau^k(P) = \frac{1}{|Clauses^k|} \sum_{C \in Clauses^k} \frac{SAT_P(C)}{maxSAT_C} \tag{23}$$

This measure checks the extent to which each clause is satisfied by P compared to the total satisfiability for all generated clauses. When this reaches 1, all clauses have been satisfied maximally, and the underlying search can stop; the set of features in P discern all considered object pairs.

Using this framework, a search amongst feature subsets can be conducted that aims to maximise the satisfiability of all generated clauses. In this work, a hill-climbing approach is adopted (see Figure 2). Initially, the k nearest neighbours are computed for each object x and stored in the list NN_x^k. The clauses are generated from these lists via generateClauses(NN_x^k,k). The process then follows the typical hill-climbing algorithm, where the addition of individual features to the current subset candidate is evaluated using the measure τ^k.

nnFDM ($\mathbb{C}$,$\mathbb{D}$,k).
$\mathbb{C}$, the set of all conditional attributes;
$\mathbb{D}$, the set of decision attributes;
k, the number of nearest neighbours to consider.

(1) $R \leftarrow \{\}$; $\tau^k_{best} = 0$;
(2) **foreach** $x \in \mathbb{U}$, calculate NN^k_x
(3) generateClauses(NN^k_x,k);
(4) **do**
(5) $T \leftarrow R$
(6) **foreach** $x \in (\mathbb{C} - R)$
(7) **if** $\tau^k(R \cup \{x\}) > \tau^k(T)$
(8) $T \leftarrow R \cup \{x\}$
(9) $\tau^k_{best} = \tau^k(T)$
(10) $R \leftarrow T$
(11) **until** $\tau^k_{best} == 1$
(12) **return** R

Figure 2: The nnFDM algorithm

The nnFRFS and nnFDM algorithms are just two of the possible ways in which nearest neighbour approaches to fuzzy-rough set feature selection can be implemented, employing the two main concepts of dependency degree and the discernibility matrices of rough set theory. However, there are many other potential extensions and applications for the proposed work and these are outlined briefly in the conclusion.

4. Feature Grouping-based Selection

One of the main drawbacks associated with conventional greedy hill-climbing approaches to discovering fuzzy-rough reducts in large datasets is that much time is wasted considering features that have strong correlation with each other. The consideration of such features is somewhat superfluous as they contain very similar information. Ultimately, evaluating all such features at each stage of the search offers no advantage. Take for example, an extreme situation where a particular dataset contains several hundred replicated features. A hill-climbing type of search will consider the addition of each of these features to the current subset candidate iteratively at each stage of the search. Obviously, such

computation is completely unnecessary. Furthermore, the later addition of any features to the subset candidate will often produce only very small improvements in the overall fuzzy-rough dependency metric [6]. The result of which is a super-reduct, i.e. the resulting subset contains superfluous features that are redundant and can be otherwise removed with no loss in dependency.

The approach proposed here (abbreviated FRFG hereafter), aims to group together similar features such that at each stage of hill-climbing, only the most promising group representative is considered for selection. This will reduce wasted computational effort, and also help to improve the final selected subset quality.

4.1. Forming Groups

An important component of the proposed approach is the identification of related features and the formation of appropriate groups. There are many measures that are useful for this task. Here, the sample correlation coefficient is used:

$$corr(a,b) = \frac{\sum_{i=1}^{|\mathbb{U}|}(a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{|\mathbb{U}|}(a_i - \overline{a})^2 \sum_{i=1}^{|\mathbb{U}|}(b_i - \overline{b})^2}} \tag{24}$$

where $a, b \in \mathbb{A}$, and $\overline{a}$ refers to the sample mean of a. This measure can be used to evaluate the degree of correlation between conditional features in order to determine groups. The sample correlation coefficient ranges from -1 to +1. In this work, the absolute value is used as a feature that is negatively correlated with another feature can also be considered to be redundant:

$$correlation(a,b) = |corr(a,b)| \tag{25}$$

The same correlation measure can be used to evaluate the correlation of conditional features with the class attribute in order to rank features within these groups. The most relevant features according to the correlation with the decision feature are therefore ranked highest in the groups. It is from these groups that the adapted hill-climbing method will select features. Redundancy

is therefore partly handled by employing groups of similar features, and relevance is considered by ranking features within groups based on their relatedness to the decision feature.

Having calculated the correlations values, groups can then formed. Here, a threshold is used to determine group membership of features. This threshold could be either a user-supplied (τ), that must be exceeded for a pair of features to be considered redundant, or could be estimated automatically:

$$\tau = 0.8(\max_{a,b \in \mathbb{C}} \{correlation(a, b)\}) \tag{26}$$

Groups are formed in the following way. For each feature f_i, the correlation with every other feature f_j is determined and the threshold (τ) applied such that if the correlation is greater than the threshold, then the feature f_j is added to the group for f_i, i.e. $F_i \leftarrow F_i \cup \{f_j\}$. Having considered all features, the result is a set of groups $F = \{F_1, F_2, ..., F_{|\mathbb{C}|}\}$. Features can be ordered within groups on the basis of their correlation with the decision feature $\mathbb{D}$, meaning that features that have greater correlation with $\mathbb{D}$ are preferable. It is important to note that as a result of this process, features can belong to more than one group.

4.2. Subset Search

Having formed the groups, the next phase of the FRFG approach is to employ the groups and their respective internal rankings in order to guide the search procedure in discovering good subsets according to a given metric. In this paper, the fuzzy-rough dependency measure is used to gauge subset quality, however any measure can be used for this purpose (including wrapper approaches). As mentioned previously, an adapted hill-climbing algorithm is used here to find the best subsets. Although there are some issues with greedy approaches (e.g. see [18]), it is still a useful search mechanism and often discovers reducts or superreducts that are usually only slightly larger than optimal. The way in which the hill-climbing search is formulated means that it is reasonably straightfor-

ward to reconfigure it for a group-based strategy. The full algorithm can be seen in Figure 3, including the required initialisation steps.

The purpose of the function preprocess(F) is to perform some initial preprocessing in order to investigate if there is any perfect correlation between features, and to remove the less relevant feature each time. This could be softened to use another threshold to remove more features (i.e. for threshold values less than 1), however this may remove useful features and prevent the algorithm from finding an optimal reduct.

For each group of features, the representative top-ranked feature is chosen and assessed by temporarily adding it to the current reduct candidate and evaluating this new subset via the metric M. In this paper, the focus is fuzzy-rough feature selection, and hence the measure used for M is the fuzzy-rough dependency degree [17]. Once a feature has been evaluated, its group members are then added to the *Avoids* set to ensure that these features are not evaluated in this iteration. The feature that produces the greatest increase in the metric is then added to the current subset and the process iterates until the stopping criterion is fulfilled. This may involve stopping when the maximum value for the measure has been reached, or to degree α, or indeed if there is no change in the measure following two successive iterations. In the fuzzy-rough case, the maximum value for a dataset can be determined prior to selection and then used as a stopping criterion.

Line (14) provides an optional further reduction in computational effort (set by the Boolean flag '`moreAvoids`') by removing all other features which appear in the group of that newly selected feature from consideration. The rationale for this step is that once a feature has been selected, the addition of any of its group members at this stage will not benefit the overall subset. There may be *some* utility in allowing the possibility of correlated group members to be added [11], but it is unlikely to have great impact on the evaluation metric. However, for flexibility, the addition of other group members of previously selected features can be permitted if this flag is set to `false`. In this work, the default setting is `true`.

In the extreme case, by setting the threshold τ=0, the algorithm then acts as a ranking approach that adds features to the reduct candidate linearly on the basis of their relevance, until the subset evaluation measure has been maximised. However, if `moreAvoids` is set to true, this behaviour will not be exhibited; instead, only the first, most relevant, feature will be chosen and then the algorithm will terminate (all other features appear in its group and are therefore removed from consideration).

The function order(F) orders the considered feature groups on the basis of their top-ranked features (i.e. most relevant), so the most promising groups are considered first. Without this, the algorithm may favour earlier features in an arbitrary fashion.

Once a feature has been added to the current subset, its group members are removed from consideration at this level. However, this does not prevent consideration of this group in future iterations. The search will stop when the stopping criterion is met. For many filter measures, a known maximum is attainable and therefore this is used to judge when to terminate the algorithm. For other measures, search can be halted when there is little or no perceived improvement in the subset quality. Also, it may be useful to stop the search somewhat prematurely by using a threshold, α, that indicates when a subset is *good enough*.

4.3. Worked Example

To illustrate the FRFG approach, an artificial example is described here. Consider a dataset with six features, some of which are highly correlated. After the initialisation steps of the algorithm, the groups formed are:

$$
\begin{aligned}
F_1 &= \{f_4, f_3, f_1\} \\
F_2 &= \{f_2\} \\
F_3 &= \{f_3, f_1\} \\
F_4 &= \{f_4, f_1, f_5\} \\
F_5 &= \{f_4, f_5\} \\
F_6 &= \{f_6\}
\end{aligned}
$$

Here, features within the groups have been ordered according to their relevance, so the left-most features are more relevant to the decision and thus are preferable to those on the right. Groups F_2 and F_6 have only one member, which indicates that features f_2 and f_6 are not strongly correlated with other features.

The hill-climbing algorithm first orders the group, say $F = \{F_4, F_3, F_1, F_5, F_2, F_6\}$ and begins the search at the first level. The first group to be considered is F_4; feature f_4 is preferable over others and is therefore added to the current (initially empty) subset R . This is then evaluated: $M(R \cup \{f_4\})$ and if it results in a better score than the current best evaluation, then feature f_4 is stored and the current best evaluation is set to $M(R \cup \{f_4\})$. The set of features which appears in group F_4 is then added to the set *Avoids* so that other group members are not evaluated in this iteration. In other words, once the main group representative has been selected, other highly correlated group members do not need to be considered. Therefore, $Avoids = \{f_1, f_4, f_5\}$ and the next feature group is considered that does not appear in *Avoids*, F_3. The highest ranked feature, f_3, is then added to the current subset and evaluated, $M(R \cup \{f_3\})$. If this value is greater than $M(R \cup \{f_4\})$, then feature f_3 replaces f_4. The set *Avoids* is then updated with the members of F_3, $Avoids = \{f_1, f_3, f_4, f_5\}$.

The next feature groups F_1 and F_5 both appear in *Avoids* and so are not considered. This means that the next considered group is F_2 (which consists of a single feature) is evaluated. Finally, the single remaining group F_6 is considered and evaluated. Having completed this, the best representative feature in this

iteration is then added to the reduct candidate R and the process iterates once more (unless the stopping criterion is not met). The *Avoids* list is reset at this level.

From this small example, it can be seen that considerable computational effort has been avoided since features f_1 and f_5 did not need to be evaluated. Note that the level of computational effort saved is governed by the group sizes, which in turn is decided by the thresholding which is used in order to form the groups. Hence, a balance must be maintained between lower thresholds (which produce larger groups, greater time saving, but potentially group fewer correlated features together) and higher thresholds (which produce smaller groups, less time saving, but features within groups are more highly correlated). In the extreme case where the threshold is set to 1, the algorithm becomes a standard hill-climber where each feature appears in exactly one group, and no time saving is made during execution. The worst-case complexity of this is $O(|\mathbb{C}|^2)$. In the other extreme, where the threshold is set to 0, all features are grouped together in ranked order and the selection process simply selects features based upon their ranking (derived form the correlation metric) until the stopping criterion is met. The worst-case complexity in this situation is linear in the number of features, $O(|\mathbb{C}|)$. Depending on the threshold value employed therefore, the actual worst-case complexity will lie somewhere between quadratic and linear for a given dataset.

5. Experimental Evaluation

This section details the experimental evaluation conducted and the results obtained for both the nnFRFS and FRFG approaches. In a series of different experiments, the proposed methods were applied to 13 datasets of different sizes, and compared with three other search methods for discovering fuzzy-rough reducts. The results presented here relate to performance in terms of quality of subsets obtained: classification accuracy and subset size, as well as execution times, and the effect of a range of threshold values for the nearest

neighbours k on the results for the nnFRFS approach.

5.1. Experimental Setup

For both approaches a total of 13 different datasets, described in Table 1, are employed. Eleven of these datasets are drawn from [12], whilst the remaining two (MIAS and DDSM) are real-world mammographic risk-assessment tasks which are related to data derived from [16].

For comparison, three other fuzzy-rough approaches for feature selection [17] are included along with three different reduct search methods: greedy hill-climbing (GHC), genetic algorithm-based search (GA), and particle swarm optimisation-based search (PSO). For the fuzzy-rough subset evaluation metric, the Łukasiewicz t-norm ($\max(x+y-1, 0)$) and the Łukasiewicz fuzzy implicator ($\min(1-x+y, 1)$) are adopted to implement the fuzzy connectives. For the similarity relation for the FRFG approach the algebraic T-norm was used ($T(x, y) = xy$). The similarity measure of eqn. (26) in [17] is also used here. It should be noted that all results that are statistically significant with respect to the greedy hill climbing fuzzy-rough feature selection approach are noted in **bold** typeface in the tables.

For the nnFDM approaches, values of 1, 3, and 5 are used for k respectively. Note that nnFRFS is not affected by the choice of value for k, as it always relies upon the closest neighbour. In the case of the FRFG approach, five different experiments are carried out by imposing different values for the threshold τ: 0.0, 0.2, 0.4, 0.6, 0.8 and 0.9. Note that for the experimentation with τ=0, `moreAvoids` is set to `false`; in this case, the algorithm will add features in order of rank to the reduct candidate until the fuzzy-rough dependency has reached its maximal value. For both nnFRFS and nnFDM, the product t-norm is used for composing similarity relations.

For the generation of classification results, two different classifier learners have been employed: JRip, a rule-based classifier [4]; and IBk [1], a nearest-neighbour classifier (with $k = 3$). Five stratified randomisations of 10-fold cross-validation were employed in generating the classification results except in the case of *lymphoma*, *leukemia*, and *colon* where the number of data objects is

small. For these three particular datasets five stratified randomisations of 3-fold cross-validation were employed. It is important to note that feature selection is performed as part of the cross-validation and each fold results in a new selection of features. A paired t-test is also used to examine the statistical significance of the generated results.

The GA search has an initial population size of 200, a maximum number of generations/iterations of 40, crossover probability of 0.6 and mutation probability of 0.033. The number of generations/iterations for PSO search was set to 40, whilst the number of particles was set to 200, with acceleration constants $c1 = 1$ and $c2 = 2$. These parameters may not be ideal for all of the datasets employed here and an optimisation phase may well result in an improvement in performance. However, such an optimisation step would need to be performed on a dataset-by-dataset basis which would involve a significant investment of effort and time and would form part of a more comprehensive future investigation. Note that GA and PSO have not been applied to the larger dimensionality datasets (*lymphoma, leukemia, colon*) as the time consumed in generating results was prohibitive, running into the many tens of hours for just a single randomisation of a single dataset.

5.2. Results: nnFRFS

Tables 2 and 3 detail the classification results for the JRip and IBk classifier learners respectively. Examining the classification results, it is clear that nnFRFS and nnFDM return very similar results to GHC. Indeed, when a paired t-test is employed to examine the statistical significance of the results generated for the proposed approaches, even though the absolute figures are slightly lower in some cases, statistically there are no inferior results. It is worth noting from Table 4, however, that the average subset sizes for nnFRFS and nnFDM are greater than GHC and the GA and PSO methods. One notable exception to this are the results for the *web* dataset, where the novel methods all return average subset sizes which are much smaller than those of all of the standard approaches.

It is in terms of execution times that both nnFRFS and nnFDM have the most to offer in terms of improvement in performance. The speed-up in performance is considerable and demonstrates that the nearest neighbour methods show potential for application to very large data. Again, the *web* dataset seems to be the exception for nnFDM at least for k=3 and k=5. (It should be noted however that the corresponding subsets discovered by GA and PSO are at least 14 times the size of those discovered by nnFDM.) This behaviour may arise as a result of the characteristics of the data itself, however, which has a large number of features and a very small number of data objects. Such datasets always present a challenge to learning algorithms regardless of the approach applied.

Dataset	Features	objects
MIAS	281	322
DDSM	281	832
web	2557	149
lymphoma	4027	96
leukemia	7130	72
colon	2001	82
cleveland	13	297
glass	9	214
heart	13	270
olitos	25	120
water2	39	390
water3	39	390
wine	13	178

Table 1: Benchmark data

One of the primary motivations behind the development of the nearest neighbour fuzzy-rough approaches detailed here was that of a reduction in computational overhead. Many of the fuzzy-rough metrics suffer in this regard when applied to larger datasets. It is clear from Table 5, that the proposed methods offer much potential in addressing this problem.

Dataset	Unred.	GHC	nnFRFS	nnFDM (k =)			GA	PSO
				1	3	5		
MIAS	63.74	60.94	58.02	58.02	61.96	60.66	64.41	53.34
DDSM	52.78	49.22	50.71	50.71	52.30	52.40	51.79	50.69
web	54.74	49.68	46.71	46.71	45.35	46.46	61.45	50.70
lymphoma	59.37	48.92	47.49	47.49	50.60	51.60	-	-
leukemia	90.67	80.05	83.29	83.29	**90.18**	**87.89**	-	-
colon	67.74	69.97	74.43	74.43	**76.62**	70.90	-	-
cleveland	54.23	54.48	54.55	54.55	54.28	54.34	54.02	54.09
glass	67.17	67.17	66.06	66.06	67.17	67.17	65.25	65.25
heart	72.96	74.15	74.22	74.44	74.67	75.41	72.30	73.85
olitos	68.50	62.83	63.33	64.00	65.67	66.83	59.33	61.17
water2	82.15	83.28	82.87	82.87	82.97	82.36	82.00	81.90
water3	82.72	81.23	81.23	81.23	81.28	82.15	78.82	78.00
wine	93.54	91.46	89.56	89.69	91.35	91.69	86.60	90.41

Table 2: nnFRFS: Classification results (%) using the JRip classifier learner

5.3. Results: FRFG

The results of the experimental evaluation are shown in Tables 6 - 10. Tables 6 – 8 detail the classification results for the J48, JRip and IBk classifier learners respectively. GHC (greedy hill-climbing), GA (genetic algorithm) and PSO (particle swarm optimisation) refer to the search technique employed in each case. Examining these results, it is clear that regardless of the value of τ, FRFG returns very similar results to GHC. Indeed, when a paired t-test is employed to examine the statistical significance of the results generated for FRFG, only those results for the *wine* dataset where τ=0.2 and 0.4 are statistically inferior to those for GHC. It is worth noting from Table 9 however, that the average subset size for these values of τ, is much smaller than for GHC indicating a trade-off between compactness of representation and model accuracy.

When FRFG is compared with the GA-based search, a similar pattern emerges. However, in this case, FRFG does not return any results which are

Dataset	Unred.	GHC	nnFRFS	nnFDM (k =)			GA	PSO
				1	3	5		
MIAS	69.57	63.29	64.10	64.10	62.54	63.83	65.40	53.48
DDSM	51.55	45.85	**51.07**	**51.07**	**51.56**	**50.69**	52.13	46.71
web	37.98	44.11	42.66	42.66	37.10	36.98	46.72	36.65
lymphoma	68.75	55.25 4	47.96	47.96	53.76	55.29	-	-
leukemia	87.5	86.97	84.64	84.64	**90.18**	86.71	-	-
colon	77.41	75.35	75.33	75.33	76.62	72.67	-	-
cleveland	56.98	52.96	**68.77**	**68.77**	55.97	56.10	53.89	53.83
glass	69.24	69.24	68.77	68.77	69.24	69.24	68.51	68.51
heart	80.96	78.15	**80.89**	**80.96**	**80.96**	**80.96**	78.15	76.96
olitos	81.00	65.67	**71.00**	70.67	**72.17**	**71.83**	66.50	72.33
water2	85.33	84.56	83.28	83.28	82.26	82.26	78.26	80.10
water3	82.97	81.23	82.00	82.00	81.08	82.05	77.44	77.23
wine	95.97	96.42	95.92	95.97	95.61	95.41	91.82	94.71

Table 3: nnFRFS: Classification results (%) using the IBk classifier learner (k=3)

statistically inferior. It is the same also for PSO, but the FRFG approach actually offers results which are statistically better than PSO for five of the datasets, most notably *wine* and *MIAS*. When considering the unreduced data, the classification results are statistically equivalent, indicating that good features are selected using the FRFG approach.

Considering the average subset size as shown in Table 9, the FRFG approach returns a range of results which seem to be similar to, or better than those of GHC. varying the value of τ generally tends to result in larger or smaller average subset size, depending on the dataset. For this comparison, the results for τ=0 are ignored as it is essentially a ranking of features, followed by the linear addition to the reduct candidate as they appear in the ranked list. For the *olitos, heart, water2* and *water3* datasets in particular, the average subset size does not seem to change significantly when $\tau \geq 0.6$. In terms of GA and PSO, the FRFG approach demonstrates a significant improvement in performance

Dataset	GHC	nnFRFS	nnFDM (k =)			GA	PSO
			1	3	5		
MIAS	6.08	13.70	13.70	17.18	19.86	9.0	7.70
DDSM	7.12	33.48	33.48	41.40	44.60	10.96	9.56
web	19.02	4.08	4.08	8.22	10.52	186.00	141.20
lymphoma	5.40	**2.00**	**2.00**	**3.22**	**4.32**	-	-
leukemia	3.80	**2.88**	**2.88**	**3.10**	**3.74**	-	-
colon	4.34	4.34	4.34	5.54	6.06	-	-
cleveland	7.64	11.08	11.10	11.80	11.82	9.0	7.70
glass	9.00	9.00	8.78	8.78	9.00	8.36	8.36
heart	7.06	10.44	10.48	10.32	10.60	7.00	7.38
olitos	5.00	7.52	7.64	8.78	9.34	5.24	5.00
water2	6.00	12.82	12.82	15.04	16.54	6.96	6.44
water3	6.08	11.42	11.42	13.40	14.70	7.00	6.50
wine	5.00	7.26	7.26	8.40	9.40	4.70	4.92

Table 4: nnFRFS: Average subset sizes

for the larger datasets: *MIAS, DDSM* and *web*. For the smaller datasets, the pattern seems to be that of equivalent or better performance (disregarding any particular value of τ).

Ostensibly, it would appear that GA-based search performs well for the *web* dataset, however if the corresponding results in Table 4 are considered, it can be seen that the average subset size is over 6.5 times that of the worst case for FRFG. The ability of FRFG to return more compact or similar sized subsets for large data whilst doing so in a much reduced execution time are encouraging. It seems that whilst FRFG offers some advantage for the smaller datasets, this varies with respect to the value of τ. This is most likely related to the process of formation of the groups. For datasets of smaller dimensionality, it may not be realistic to form reasonable groups based on higher values of τ as there may be lower levels of overall redundancy.

The approaches and ideas described in this paper offer some new directions

Dataset	GHC	nnFRFS	nnFDM (k =)			GA	PSO
			1	3	5		
MIAS	12.04	**1.05**	**1.07**	**1.67**	**2.99**	3.11	22.60
DDSM	110.44	**7.12**	**6.97**	**15.58**	**26.13**	23.94	173.93
web	98.42	**1.63**	**2.33**	**3.70**	**5.72**	3.51	24.07
lymphoma	584.3	**3.90**	**4.63**	**3.22**	**4.32**	-	-
leukemia	497.30	**3.92**	**6.82**	**10.32**	**14.10**	-	-
colon	9.28	**0.87**	**1.10**	**1.87**	**2.91**	-	-
cleveland	0.39	**0.037**	**0.05**	**0.06**	**0.07**	16.20	3.83
glass	0.14	**0.02**	**0.03**	**0.04**	**0.05**	1.55	1.08
heart	0.30	**0.03**	**0.04**	**0.05**	**0.06**	14.48	3.46
olitos	0.11	**0.02**	**0.03**	**0.04**	**0.05**	2.36	1.26
water2	2.16	**0.13**	**0.14**	**0.2**	**0.27**	20.14	19.71
water3	2.17	**0.14**	**0.15**	**0.21**	**0.28**	19.57	17.25
wine	0.11	**0.02**	**0.03**	**0.04**	**0.06**	7.55	1.29

Table 5: nnFRFS: Average execution times per fold (sec.)

for further development. In particular, (and as mentioned previously) the FRFG algorithm is a general approach, and it is not limited to the use of the fuzzy-rough set subset evaluator and indeed any metric can be used for this purpose. As such, it would be interesting to investigate the advantages for other metrics, particularly those which perform well but may not scale-up for larger datasets. One of the other aspects that may provide some additional potential for the approach is an in-depth investigation of the effects of the choice of value for the parameter τ. This may provide some insight into how the value can be selected automatically or indeed derived from the data.

Another important factor is how groups are formed; in the present approach, the sample correlation is used as the basis for group membership. Although this means that the number of groups is initially the same as the number of features, the impact of this is reduced by the use of `moreAvoids` and the appropriate choice of parameter value for τ. This may still pose a problem for very large

Dataset	Unred.	GHC	FRFG						GA	PSO
			$\tau =$							
			0.0	0.2	0.4	0.6	0.8	0.9		
MIAS	66.72	60.11	57.22	61.98	62.99	61.42	61.63	59.26	61.88	52.67
DDSM	50.16	46.40	44.24	50.69	46.86	47.20	45.48	46.43	48.71	47.39
web	56.32	50.32	**55.70**	51.43	51.30	51.40	51.69	50.74	56.49	50.17
lymphoma	70.45	68.80	58.78	62.53	60.02	54.60	64.64	67.36	-	-
leukemia	84.72	89.98	**91.86**	**91.04**	**90.18**	**83.89**	**89.04**	**89.36**	-	-
colon	83.87	81.62	**83.24**	79.43	72.14	71.71	70.19	69.14	-	-
cleveland	54.03	51.61	54.96	54.03	52.35	51.54	51.54	51.54	52.68	53.31
glass	67.54	67.54	67.54	62.25	66.87	66.31	66.02	67.54	67.44	67.44
heart	75.56	74.74	76.74	77.11	77.11	74.15	74.15	74.15	75.48	76.37
olitos	66.67	60.67	60.50	63.00	62.00	61.83	60.67	60.67	57.67	65.67
water2	82.56	83.49	83.44	81.95	81.74	82.10	82.41	83.69	81.18	81.44
water3	82.67	80.92	81.28	81.13	79.59	81.08	79.79	80.62	76.82	77.95
wine	93.82	95.39	93.82	79.54	87.29	91.39	95.05	95.27	88.73	90.86

Table 6: FRFG: Classification results (%) using the J48 classifier learner

datasets, however, so an alternative feature clustering scheme could be adopted in order to ensure quick clustering and small group sizes. One such clustering mechanism is presented in [14], which employs a rough set discernibility-based attribute similarity measure for identifying interchangeable groups of attributes. This could be extended to fuzzy-rough discernibility and utilised in the present work, resulting in a true fuzzy-rough approach to group-based feature selection.

6. Conclusion

Two approaches which help in alleviating computational effort for feature selection based upon fuzzy-rough sets have been presented in this paper. They are based upon two different ideas related to tackling the problems associated with larger data. The first calculates nearest data object neighbours prior to

Dataset	Unred.	GHC	FRFG $\tau =$						GA	PSO
			0.0	0.2	0.4	0.6	0.8	0.9		
MIAS	63.74	60.94	57.09	63.10	60.84	61.74	61.19	60.26	64.41	53.34
DDSM	52.78	49.22	48.88	51.14	50.21	49.65	47.77	48.73	51.79	50.69
web	54.74	49.68	**55.94**	51.40	51.57	50.22	**52.66**	51.96	61.45	50.70
lymphoma	59.37	48.92	51.13	**53.93**	**56.24**	**54.38**	**64.71**	**64.91**	-	-
leukemia	90.67	80.05	**91.54**	**91.89**	**90.25**	82.89	**86.64**	**88.29**	-	-
colon	67.74	69.97	**81.86**	**78.76**	**76.14**	72.38	69.14	67.38	-	-
cleveland	54.23	54.48	55.22	53.22	54.41	54.48	54.48	54.48	54.02	54.09
glass	67.17	67.17	67.17	60.56	66.68	65.05	64.95	67.17	65.25	65.25
heart	72.96	74.15	74.15	74.44	74.96	73.93	73.93	73.93	72.30	73.85
olitos	68.50	62.83	60.00	61.67	59.00	59.50	59.67	59.67	59.33	61.17
water2	82.15	83.28	82.87	82.15	82.05	82.21	82.97	83.69	82.00	81.90
water3	82.72	81.23	82.56	81.18	80.36	81.28	80.87	81.74	78.82	78.00
wine	93.54	91.46	92.69	76.61	86.72	90.33	93.25	93.38	86.60	90.41

Table 7: FRFG: Classification results (%) using the JRip classifier learner

the search and then uses only these neighbours for the subsequent fuzzy-rough dependency calculations. The time complexity therefore is essentially an order of magnitude smaller for the number of data objects. The second approach is an attempt to tackle the problem of larger data from the perspective of large dimensionality, and groups and ranks features in a preprocessing step prior to selection.

For nnFRFS, the results detailed in the previous section show that the average subset sizes are slightly larger than those of existing approaches, but for FRFG the subset sizes are comparable to those of GHC. What is clear from the experimental evaluation is the level of reduction in execution times for both approaches. This suggests that approaches such as those detailed in this paper offer a number of possible avenues of exploration which would offer improvements in the performance in terms of subset size, whilst retaining the saving in

Dataset	Unred.	GHC	FRFG						GA	PSO
			$\tau =$							
			0.0	0.2	0.4	0.6	0.8	0.9		
MIAS	69.57	63.29	58.72	63.38	62.64	63.16	61.38	58.61	65.40	53.48
DDSM	51.55	45.85	45.34	46.97	47.63	45.39	45.85	46.00	52.13	46.71
web	37.98	44.11	39.20	48.83	45.08	45.32	42.77	41.07	46.72	36.65
lymphoma	68.75	55.25	56.78	**63.20**	**61.47**	55.04	**71.31**	**70.93**	-	-
leukemia	87.50	86.97	**91.54**	87.43	**90.54**	85.21	87.04	89.36	-	-
colon	77.41	75.35	**81.76**	80.33	73.14	71.52	70.76	74.33	-	-
cleveland	56.98	52.96	56.91	50.79	54.77	52.96	52.96	52.96	53.89	53.83
glass	69.24	69.24	69.24	59.87	63.28	68.23	68.52	69.24	68.51	68.51
heart	80.96	78.15	81.11	75.85	79.85	77.56	77.56	77.56	78.15	76.96
olitos	81.00	65.67	65.67	66.33	67.67	65.67	66.83	66.83	66.50	72.33
water2	85.33	84.56	87.08	84.97	82.21	83.49	84.77	85.33	78.26	80.10
water3	82.97	81.23	86.36	80.92	81.54	82.51	80.36	80.92	77.44	77.23
wine	95.97	96.42	96.96	73.75	90.21	92.59	95.15	95.05	91.82	94.71

Table 8: FRFG: Classification results (%) using the IBk classifier learner (k=3)

computational effort. For example, the use of propositional satisfiability techniques [18] to find the smallest reducts in the clauses generated using nnFDM, or applying the approaches to unsupervised FS, and improving the efficiency of recent fuzzy-rough object/object selection methods, etc. In addition, an extension of the proposed approaches to a distributed environment such as that described in [2] may also be an interesting proposal. The combination of either of the approaches (nnFRFS/FRFG) with other techniques in order to form hybrid preprocessors may offer ways of further reducing computational overhead.

The experimental evaluation in this paper features at least three large datasets, however it would be interesting to apply nnFRFS and FRFG to data in the order of thousands of features and objects; this would also form the basis for a more comprehensive investigation.

Dataset	GHC	FRFG						GA	PSO
		$\tau =$							
		0.0	0.2	0.4	0.6	0.8	0.9		
MIAS	6.08	19.02	**4.50**	6.40	6.28	6.24	6.22	9.0	7.70
DDSM	7.12	34.26	**4.94**	7.44	7.32	7.10	7.16	10.96	9.56
web	19.02	496.40	28.42	22.00	19.64	19.40	19.06	186.00	141.20
lymphoma	5.40	**1.98**	5.10	4.86	**4.48**	**4.04**	**4.02**	-	-
leukemia	3.80	**1.98**	3.62	3.56	3.34	3.30	3.22	-	-
colon	4.34	**2.00**	5.24	4.38	4.04	4.00	4.00	-	-
cleveland	7.64	12.08	**5.52**	6.40	6.28	7.64	6.22	9.0	7.70
glass	9.00	9.00	**3.16**	**5.02**	**8.00**	**8.12**	9.00	8.36	8.36
heart	7.06	11.00	**5.24**	8.06	7.06	7.06	7.06	7.00	7.38
olitos	5.00	6.38	5.52	5.04	5.00	5.00	5.00	5.24	5.00
water2	6.00	6.98	6.86	6.10	6.04	6.00	6.00	6.96	6.44
water3	6.08	7.80	6.76	6.16	6.04	6.00	6.00	7.00	6.50
wine	5.00	5.40	**1.80**	4.88	4.94	4.98	5.00	4.70	4.92

Table 9: FRFG: Average subset sizes

Acknowledgment

Neil Mac Parthaláin would like to acknowledge the financial support for this research through NISCHR (*National Institute for Social Care and Health Research)* Wales, Grant reference: RFS-12-37.

[1] D. Aha, D. Kibler. object-based learning algorithms, Machine Learning, vol.6, pp. 37–66, 1991.

[2] H. Asfoor, R. Srinivasan, G. Vasudevan, N. Verbiest, C. Cornelis, M. Tolentino, A. Teredesai, M. De Cock, Computing Fuzzy Rough Approximations in Large Scale Information Systems in: Proceedings of 2nd Workshop on Scalable Machine Learning: Theory and Applications (workshop at IEEE BigData 2014), pp. 9–16, 2014.

Dataset	GHC	FRFG						GA	PSO
		$\tau =$							
		0.0	0.2	0.4	0.6	0.8	0.9		
MIAS	12.04	**0.96**	**0.57**	**0.8**	**1.09**	**1.61**	**2.76**	3.11	22.60
DDSM	110.44	**4.17**	**2.32**	**4.32**	**8.26**	**12.85**	**25.09**	23.94	173.93
web	98.42	**11.60**	**13.29**	**18.53**	**37.21**	**67.24**	**81.57**	3.51	24.07
lymphoma	584.3	**10.67**	**10.98**	**11.01**	**14.84**	**15.54**	**16.00**	-	-
leukemia	497.30	**24.26**	**24.35**	**23.42**	**25.11**	**28.39**	**28.55**	-	-
colon	9.28	**1.36**	**1.44**	**1.58**	**1.60**	**1.86**	**2.57**	-	-
cleveland	0.39	**0.13**	**0.14**	0.40	0.45	0.43	0.45	16.20	3.83
glass	0.14	**0.07**	**0.05**	**0.08**	0.15	0.15	0.19	1.55	1.08
heart	0.30	**0.11**	**0.11**	0.31	0.36	0.34	0.35	14.48	3.46
olitos	0.11	**0.05**	**0.06**	**0.09**	0.12	0.14	0.15	2.36	1.26
water2	2.16	**0.18**	**0.40**	**0.69**	**0.98**	**1.48**	**1.87**	20.14	19.71
water3	2.17	**0.20**	**0.41**	**0.75**	**1.03**	**1.5**	**1.87**	19.57	17.25
wine	0.11	**0.04**	**0.04**	**0.08**	0.12	0.13	0.16	7.55	1.29

Table 10: FRFG: Average execution times per fold (sec.)

[3] R. Bekkerman, M. Bilenko, and J. Langford, (eds). Scaling up machine learning: Parallel and distributed approaches. Cambridge University Press, 2011.

[4] W.W. Cohen. Fast effective rule induction, Proceedings of the 12th International Conference on Machine Learning, pp. 115–123, 1995.

[5] C. Cornelis, N. Verbiest and R. Jensen. Ordered Weighted Average Based Fuzzy Rough Sets. Proceedings of the 5th International Conference on Rough Sets and Knowledge Technology (RSKT2010), pp. 78-85, 2010.

[6] C. Cornelis, R. Jensen, G. Hurtado Martín, D. Ślęzak, Attribute Selection with Fuzzy Decision Reducts, Information Sciences, vol. 180, no. 2, pp. 209–224, 2010.

[7] M. Cox and D. Ellsworth. "Managing big data for scientific visualization." ACM Siggraph. vol. 97, 1997.

[8] D. Dubois and H. Prade, Putting Rough Sets and Fuzzy Sets Together, *in* Intelligent Decision Support, pp. 203–232, 1992.

[9] J. Fan, R. Samworth, and Y. Wu. Ultrahigh dimensional feature selection: beyond the linear model. Journal of Machine Learning Research, vol. 10, pp.2013–2038, 2009.

[10] M. Ferguson, Architecting A Big Data Platform for Analytics. A Whitepaper Prepared for IBM, 2012.

[11] I. Guyon and A. Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003.

[12] A. Frank and A. Asuncion. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.

[13] Q. Hu, L. Zhang, S. An, D. Zhang, and D. Yu, "On Robust Fuzzy Rough Set Models," IEEE Transactions on Fuzzy Systems, vol. 20, no. 4, pp. 636–651, 2012.

[14] A. Janusz and D. Ślęzak. Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data, 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), pp.295–302, 2012.

[15] J. Manyika, M. Chui, B. Brown, J. Bughin, R.Dobbs, C. Roxburgh, and A.H. Byers, Big data: The next frontier for innovation, competition. and productivity. Technical Report, McKinsey Global Institute, 2011.

[16] A. Oliver, J. Freixenet, R. Marti, J. Pont, E. Perez, E.R.E. Denton, R. Zwiggelaar. A Novel Breast Tissue Density Classification Methodology.

IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 1, pp. 55–65, 2008.

[17] R. Jensen and Q. Shen. New Approaches to Fuzzy-Rough Feature Selection, IEEE Transactions on Fuzzy Systems, vol. 17, no. 4, pp. 824–838, 2009.

[18] R. Jensen, A. Tuson, and Q. Shen. Finding rough and fuzzy-rough set reducts with SAT, Information Sciences, vol. 255, pp. 100–120, 2014.

[19] R. Jensen and N. Mac Parthaláin. Nearest Neighbour-Based Fuzzy-Rough Feature Selection. Lecture Notes in Computer Science Volume 8536, pp. 35–46, 2014.

[20] R. Jensen, N. Mac Parthaláin and C. Cornelis. Feature Grouping-Based Fuzzy-Rough Feature Selection. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE14), pp. 1488–1495, 2014.

[21] Z. Pawlak. Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishing, 1991.

[22] Y. Qu, Q. Shen, N. Mac Parthaláin, C. Shang, and W. Wu, Fuzzy similarity-based nearest-neighbour classification as alternatives to their fuzzy-rough parallels, International Journal of Approximate Reasoning, vol. 54, no. 1, pp. 184-195, 2012.

[23] A.M. Radzikowska and E.E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems, vol. 126, no. 2, pp. 137–155, 2002.

[24] S. Singh, J. Kubica, S. Larsen, and D. Sorokina, Parallel Large Scale Feature Selection for Logistic Regression. In Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 1172–1183, 2009.

[25] M. Tan, I.W. Tsang, and L. Wang. Towards Ultrahigh Dimensional Feature Selection for Big Data, Journal of Machine Learning Research, vol. 15, pp. 1371–1429, 2014.

[26] J. Zhang, T. Li, D. Ruan, Z. Gao, and C. Zhao. A parallel method for computing rough set approximations. Information Sciences, vol. 194, pp. 209–223, 2012.

FRFG($\mathbb{C}$, $\mathbb{D}$, M,τ, `moreAvoids`).

$\mathbb{C}$, the set of conditional features;

$\mathbb{D}$, the set of decision features;

M, subset evaluation measure;

τ, the group-forming threshold;

`moreAvoids`, Boolean variable

(1) $R \leftarrow \emptyset$; $F \leftarrow$ formGroups($\mathbb{C}, \tau$)
(2) $F \leftarrow$ rankWithinGroups($\mathbb{D}$,F)
(3) preprocess(F); $F \leftarrow$ order(F); $AlwaysAv \leftarrow \emptyset$
(4) **while** (stopping criterion not met)
(5) $\quad Avoids \leftarrow AlwaysAv$
(6) $\quad$ bestF $\leftarrow \emptyset$; bestEval $= 0$
(7) $\quad$ **foreach** $a \in (\mathbb{C} - R - Avoids)$
(8) $\quad\quad a \leftarrow$ highestRankedFeature(F_a)
(9) $\quad\quad T \leftarrow R \cup \{a\}$
(10) $\quad\quad$ **if** $(M(T) >$bestEval)
(11) $\quad\quad\quad$ bestF $= a$; bestEval $= M(T)$
(12) $\quad\quad Avoids \leftarrow Avoids \cup F_a$
(13) $\quad R \leftarrow R \cup$ bestF
(14) $\quad$ **if** (`moreAvoids`)
(15) $\quad\quad AlwaysAv \leftarrow AlwaysAv \cup F_{bestF}$
(16) **output** R

Figure 3: The feature grouping algorithm